EXPERIMENT NO.07

(1)BFS

C grapghbfs.c > main()

```c
#include <stdio.h>

int a[20][20], q[20], visited[20], n, f = -1, r = -1;

void bfs(int v) {
    int i;
    for (i = 0; i < n; i++) {
        if (a[v][i] != 0 && visited[i] == 0) {
            r = r + 1;
            q[r] = i;
            visited[i] = 1;
            printf("%d ", i);
        }
    }
    f = f + 1;
    if (f <= r) {
        bfs(q[f]);
    }
}

int main() {
    int v, i, j;
    printf("\nEnter number of vertices: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        visited[i] = 0;
    }
    printf("\nEnter graph data in matrix form:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    printf("\nEnter the starting vertex: ");
    scanf("%d", &v);
    f = 0;
    r = 0;
```

grapghbfs.c > breadth_first_search(int)

```c
void bfs(int v) {
    f = f + 1;
    if (f <= r) {
        bfs(q[f]);
    }
}

int main() {
    int v, i, j;
    printf("\nEnter number of vertices: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        visited[i] = 0;
    }
    printf("\nEnter graph data in matrix form:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    printf("\nEnter the starting vertex: ");
    scanf("%d", &v);
    f = 0;
    r = 0;
    q[r] = v;
    visited[v] = 1;
    printf("%d ", v);
    bfs(v);
    if (f == n - 1) {
        printf("\nBFS not possible");
    }
    printf("\n");
    return 0;
}
```

OUTPUT:

```
Enter number of vertices: 3

Enter graph data in matrix form:
1 0 1
0 1 0
1 1 0

Enter the starting vertex: 2
2 0 1
```

(2)DFS

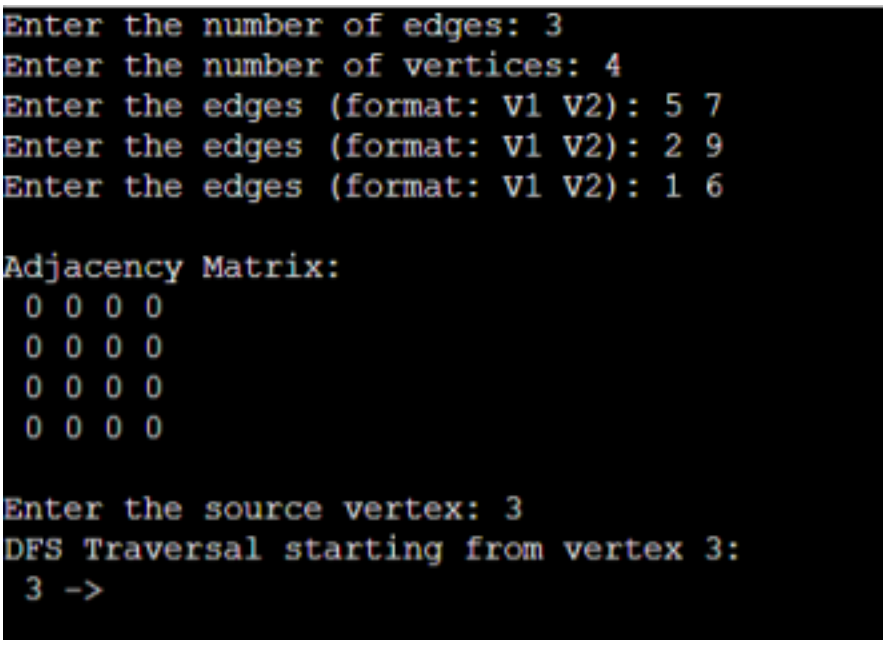
grapghDFS.c > DFS(int)

```c
#include <stdio.h>
#include <stdlib.h>

int G[20][20], visited[20], v;

void DFS(int t) {
    int j;
    visited[t] = 1;
    printf(" %d ->", t + 1);
    for (j = 0; j < v; j++) {
        if (G[t][j] == 1 && visited[j] == 0) {
            DFS(j);
        }
    }
}

int main() {
    int i, j, e, v1, v2, source;

    printf("Enter the number of edges: ");
    scanf("%d", &e);

    printf("Enter the number of vertices: ");
    scanf("%d", &v);

    for (i = 0; i < v; i++) {
        for (j = 0; j < v; j++) {
            G[i][j] = 0;
        }
        visited[i] = 0;
    }

    for (i = 0; i < e; i++) {
        printf("Enter the edges (format: V1 V2): ");
        scanf("%d %d", &v1, &v2);
        G[v1 - 1][v2 - 1] = 1;
    }
```

0 C/C++ Runner: Debug Session (dsa)

```
grapghDFS.c > DFS(int)
int main() {
                G[i][j] = 0;
            }
            visited[i] = 0;
        }

        for (i = 0; i < e; i++) {
            printf("Enter the edges (format: V1 V2): ");
            scanf("%d %d", &v1, &v2);
            G[v1 - 1][v2 - 1] = 1;
        }

        printf("\nAdjacency Matrix:\n");
        for (i = 0; i < v; i++) {
            for (j = 0; j < v; j++) {
                printf(" %d", G[i][j]);
            }
            printf("\n");
        }

        printf("\nEnter the source vertex: ");
        scanf("%d", &source);

        printf("DFS Traversal starting from vertex %d:\n", source);
        DFS(source - 1);

        return 0;
    }
```

OUTPUT:



STUDENT NAME:FARID IQRA AFNAN
CLASS/DIV:SY-IT-A
ROLL NO.25